Primer Trabajo Práctico

14 de Abril de 2010 — Algoritmos y Estructuras de Datos III

Integrante	LU	Correo electrónico
Bianchi, Mariano	92/08	bianchi-mariano@hotmail.com
Brusco, Pablo	527/08	pablo.brusco@gmail.com
Di Pietro, Carlos Augusto Lyon	126/08	cdipietro@dc.uba.ar

Facultad de Ciencias Exactas y Naturales

Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

http://www.fcen.uba.ar

Índice

Ejercicio 1

Introducción

El primer problema del presente trabajo consistió en la implementación de un algoritmo capaz de dar solución a la ecuación

$$b^n \ mod \ (n) \tag{1}$$

haciendo uso de alguna de las técnicas algorítmicas aprendidas hasta el momento en la materia. Asímismo, la consigna dictaba que la complejidad final del algoritmo debería ser menor a $O(n)$. En pos de cumplimentar lo pedido se decidió usar la técnica de *Dividir & Conquistar*[1] para desarrollar el algoritmo. Esta técnica se caracteriza principalmente en dividir la instancia de un problema en instancias más pequeñas, atacar cada una de ellas por separado y resolverlas, para finalmente juntar sus resultados y aséi producir el resultado final.

Detalles de implementación

La primera solución que se piensa casi de manera intuitiva es la de mutliplicar n veces el número b y luego hallar el resto de dividir ese resultado por n.

$$\underbrace{b.b.b \ldots b.b.b}_{n \ veces} \ mod \ (n) = b^n \ mod \ (n) \tag{2}$$

Sin embargo, la complejidad ese algoritmo es $O(n)$, ya que se realizan n multipicaciones y 1 división, razón por lo cual no cumple con lo pedido en la consigna.
No obstante, la idea anterior conduce a otra forma de encarar el problema. Veamos de qué se trata.

Sabemos que $b^n \ mod \ (n)$ es el resto de dividir b^n por n. Llamemos x a ese resto. Luego:

$$x = b^n \ mod \ (n) \ , \quad con \ 0 \leq x < n \tag{3}$$

ya que el Algoritmo de División de Números Enteros asegura que el resto existe, es único y es un valor entre 0 y n.

Inmediatamente de lo anterior, por la definición de congruencia, se desprende que:

$$b^n \ \equiv \ x \ (n) \tag{4}$$

Luego, una de las propiedades de congruencias, nos asegura que:

$$a \equiv p(n) \ , \ b \equiv q(n) \implies a.b \equiv p.q(n) \ , \quad \forall \ a, b, p, q, n \in \mathbf{Z} \tag{5}$$

por lo que, tomando en particular, $a = b$ sigue que:

$$b \equiv \ p(n) \implies b.b \equiv p.p \ \Leftrightarrow \ b^2 = p^2(n) \tag{6}$$

Sea ahora k el resto de la division de p^2 por n; luego:

$$p^n \ \equiv \ k \ (n) \ , \quad con \ 0 \leq k < n \tag{7}$$

[1] Poner alguna referencia en donde se explique esta técnica

Por lo tanto, de 6 y 7 se obtiene usando la propiedad transitiva de congruencias que:

$$b^2 \equiv p^2(n)\ ,\ p^2 \equiv k(n) \implies b^2 \equiv k(n) \tag{8}$$

donde por 7 sabíamos que k es un número entre 0 y n.

De este modo, podemos aplicar el resultado de 8 en 4, obteniéndose:

$$b^n = \underbrace{b.b.b \ldots b.b.b}_{n\ veces} \equiv x(n) b^n = \underbrace{b^2.b^2 \ldots b^2.b^2}_{n/2\ veces} \equiv x(n) b^n = \underbrace{k.k \ldots k.k}_{n/2\ veces} \equiv x(n) \tag{9}$$